Geoinformatics 2004
Proc. 12th Int. Conf. on Geoinformatics − Geospatial Information Research: Bridging the Pacific and Atlantic
University of Gävle, Sweden, 7-9 June 2004

A NETWORK COMMON DATA FORM (NETCDF) UTILITY FOR EFFICIENT ENVIRONMENTAL DATA PROCESSING AND VISUALIZATION

Jinxun Liu[1,3], Jing M. Chen[1], David T. Price[2], and Shuguang Liu[3]
[1]Department of Geography and Program in Planning, University of Toronto, 100 St. George St., Room 5047, Toronto, Ontario, Canada M5S 3G3
[2]Natural Resources Canada, Canadian Forest Service, 5320-122nd St., Edmonton, AB, Canada T6H 3S5
[3]SAIC, USGS EROS Data Center, Sioux Falls, SD 57198, USA

Abstract

NetCDF is a data sharing protocol and library that is commonly used in large-scale atmospheric and environmental data archiving and modelling. Most current NetCDF-related utilities are for Unix and Linux platforms while only a few, mostly commercial, are for the Microsoft Windows platform. The NetCDF tool described here was built as a standard executable as well as a COM (component object model) application using Borland C++ Builder in the Windows environment. This NetCDF tool has three major components: 1) Data conversion, 2) Data visualization, and 3) Modeling interface. The data conversion part is designed to convert binary raw data to NetCDF data and vise versa. It can process six data types (unsigned char, signed char, short, int, float, double) and three spatial data format (BIP, BIL, BSQ). The visualization part is designed for displaying grid map series and displaying temporal trend curves of individual map pixels. It can also draw simple map legend. The modeling interface is designed as a COM. COM is intended for cross-platform application development and it is a powerful technology that enhances the reuse of application components. Environmental model developers from different modeling environments, such as JAVA, VISUAL FORTRAN, VISUAL BASIC, VISUAL C++, and DELPHI, can reuse this NetCDF COM object in their model to read and write NetCDF data. The data conversion and presentation parts are also installed with COM interfaces so they can be also called from within another application. An example of using Microsoft Power Point to call this NetCDF tool for displaying map animations is shown.

INTRODUCTION

The common tasks of processing and visualizing spatial-temporal data usually involve the use of commercial remote sensing or GIS software. But linking an environmental model to remote sensing or GIS software at program code level may not be so easy. Therefore specific spatial-temporal data processing tools would be welcome. The Network Common Data Form (NetCDF) is such a non-commercial tool. NetCDF has been developed since the early 90s under a program named UNIDATA supported by the American National Science Foundation. In fact it is a data sharing protocol (and also a library) for storing and retrieving scientific data in self-describing, platform-independent files (Rew and Davis, 1990, 1997). NetCDF is primarily intended for atmospheric related research and now it becomes a commonly accepted data sharing and processing means. Many research programs use NetCDF as a data interface for their models. Examples include climate change (Benestad, 2000) and some large-scale terrestrial ecosystem simulators, such as

IBIS (Foely et al., 1996), SiB2 (Berry et al., 1997), VEMAP (Kittel et al., 1997), and the newer version of CENTURY model developed by Parton et al. (1987). However, most current NetCDF tools are for the Unix and Linux systems while NetCDF tools for Windows are few and usually commercial. Because many people nowadays do spatial-temporal modelling work under the Windows environment, it is beneficial to provide an integrated NetCDF tool for the Windows environment. This paper presents a Windows NetCDF utility (named as NCWin) that includes a NetCDF modeling interface, a data conversion component and a data visualization component.

The NCWin is developed not only as a standard executable, but also a Component Object Model (COM) mainly for the purpose of model reuse. The COM and the related Distributed COM (DCOM) and COM+ are the Microsoft technological proposals for developing component-based software. It is a specification for defining an object that can be used across applications and languages (Rogerson, 1997; Reisdorph, 1999). So, COM based NCWin can be reused or invoked by other applications or modelling programs. COM in environmental modelling has already been discussed (Smith, 1997; Potter et al., 2000; Liu et al., 2002). Potter et al. (2000) pointed out that an ideal forest ecosystem management decision support system takes advantage of the combined capability of many available systems working together and that COM is currently the most suitable technology for model reuse. NCWin potentially can be incorporated into various ecosystem models that might facilitate data processing and promote data sharing. Basically, the descriptions here apply to general COM based environmental modelling.

COM ADVANTAGES AND THE NCWIN FRAMEWORK

COM as reusable model components

COMs are relatively independent and fully functional binaries that can do certain jobs as standard executables do. For example, it can be a sub-model of a modelling system, or a data processing utility that deals with model input and output. COMs are language independent, which means they can be used by various programming languages. The concept of using a COM is that a client program can activate a COM server and establish communications with it (Figure 1).

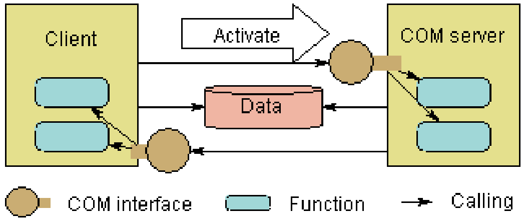

Figure 1: Client and Server communication through a COM interface.

The connection uses the interfaces defined by the COM. When the server is activated it exposes various interfaces for the client to use. Because the client already knows what interfaces the server has and knows how to call the interface functions, it can pass data and parameters to the server. The server will do the required job using its internal functions and output the results. The server can also send out signals telling the client that certain jobs have been finished (In such conditions, the client needs to install COM interfaces in order to get feedbacks from the COM server).

Using COM is different than using executable. At least, calling an executable is usually invoking the whole of the application and the user has to go through the processes fixed by the application, which is usually slow in speed and consumes considerable computer memory. Calling a COM is like using a back door of a program, and the user can directly focus on the required part. An example of COM product is the ESRI MapObject. Unlike the ArcView application, MapObject is provided for GIS modelers who want to build customized GIS applications by themselves. For instance, the user can call MapObject's functions for map display, or zooming, directly in his own program. Another common example is Microsoft Excel, which is an executable, but also a COM server that bears many COM interfaces and functions. When a user wants to put some data and draw a chart into an Excel file, directly from his own program, he can use the COM technology.

Sharing a COM is also different than sharing the program source code. Most environmental modelers are hesitant in sharing their source codes. But with COM they can share their models by only providing the functionality, without source codes.

The complexity of COM processes includes querying interfaces, passing parameters and data references, firing and sinking events, thread synchronizing, and even cross network proxy. Much information about COM can be found from books and the Internet (http://www.microsoft.com/com/).

General description of NCWin

The NCWin is designed to do three major tasks: modeling, data conversion, and data visualization (Figure 2).

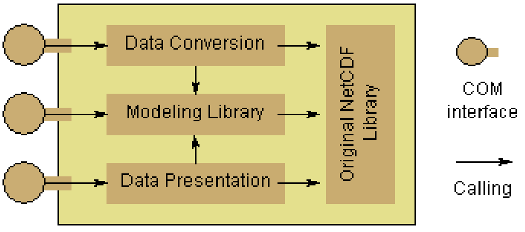

Figure 2: NCWin COM components.

NCWin modeling component

This component makes it possible that a client model can create, read and write NetCDF files while the client model is running. This task is done by using COM-based NetCDF functions that call functions of the original NetCDF library provided by UNIDATA. Currently there are seven COM NetCDF functions for the client to use. Most of them contain multiple original NetCDF functions:

- *NCCreate* – Create a NetCDF file with all dimensions defined and assigned;
- *NCClose* – Close an opened NetCDF file;
- *NCVariable* – Define a new NetCDF variable with dimensions;
- *NCAttributes* – Add proper attributes to dimensions and variables;
- *NCRead* – Read data from a NetCDF file;
- *NCWrite* – Write data to a NetCDF file;
- *GetNCInfo* – Get all information from a NetCDF file.

Although there are about 150 functions defined by the original NetCDF library that give maximum control of NetCDF files, the above COM functions are simple and practical to deal with normal NetCDF file processing in a user's program. It is not difficult to convert all the original NetCDF functions into the COM format so that the same COM library can be used with different programming languages.

Data conversion component

This component mainly converts raw data to and from NetCDF data. Unlike the modelling interface, it is basically for use outside a model simulation process. Raw data here means temporal-spatial data without additional information (such as a header). If the data dimensions and data types are known, the conversion is quite simple. Major conversions of this component include:

Data type and Geo-format Conversion – The conversion component can convert data among six types (unsigned char, signed char, short, int, float, double). A simple mechanism for checking the file size, dimension size and date type size secures the conversion process, and will notify the user when there is a mis-match on the size requirement. The geo-format (BIP, BIL, BSQ) conversion rearranges binary data according to their dimensions. But a NetCDF file is "always" in BSQ format in this COM, i.e. all NetCDF map series are stored frame by frame in computer's physical memory. Functions in this part include Dat2NC, NC2Dat, and Dat2Dat. Some special ASCII file conversions are also included.

Data Aggregation – NetCDF map can be scaled up in order to reduce the file size. In this process a new pixel will be an area-weighted product from the original pixels. For example, if we scale a map to 10 (merge 10 units along both latitude dimension and longitude dimension), then one new pixel will contain 100 old pixels. Suppose we are dealing with a land mask map, if the original land pixels are more than half of the total 100 pixels, this new pixel will be a land pixel. The percentage of actual land area to the total area is also stored as sub-pixel information for potential use.

Other capabilities are provided such as adding new variables to an existing NetCDF file, clipping a subset of data, converting 3-D data to 4-D data, adding missing values, scale factor, offset, and adding maximum and minimum values.

The whole data conversion component is a COM unit that contains a dialog window. The related conversion functions can also be called directly through their own COM functions that do not require opening the dialog window.

Data visualization component

This is a standalone module with two displaying windows:

Map Display – Display NetCDF map series with an optional legend. It shows the cursor location on the map and the value of a map pixel.

Curve Display – Display temporal trends of individual map pixels by double clicking a pixel on a NetCDF map.

Some details about NCWin are provided in the following section.

NCWIN USAGE

COM modelling interface

The NCWin library was created at the time NCWin COM was built, which contains all the information of NCWin. If a client model uses the NCWin library, it will be able to create a NCWin object and use it when the client model is run. The client model knows what services NCWin can provide and what parameters are needed for the services. For example, if the client wants to create a NC file and write data to it, the client first needs to prepare related information such as file name, variable name, dimension size, variable value vector, etc. Then the client can finish the task by calling NCWin's COM functions:

```
ClientFunction(){
   NCCreate (file name, dimensions, …);
   NCVariable (variable name, variable dimmensions, …);
   NCAttributes (variable name, attribute, …);
   NCWrite (variable name, variable value, …);
   NCClose (file name);
}
```

The COM functions in turn call original NetCDF functions. For example, the NCCreate (…) actually includes multiple calls:

```
NCCreate(…){
   nc_create(…);   //create new NC file with a given name;
   def_dim(…);     //define dimensions, usually 4D;
   nc_enddef(…);  //change file to data mode so as to read or write data
   put_var(…);     //assign values to each dimension;
}
```

Since the parameters are already provided when the client calls the COM functions, the user does not need to worry about calling the original NetCDF functions. This is a way of simplifying the NetCDF usage. The way of accessing a COM object in order to use the COM functions vary depending on different programming languages. A small example will be given later.

Data conversion dialog window

Figure 3 shows a NetCDF file conversion dialog window. Most of the background functions are from the modelling interface and original NetCDF library. The above portion of the window is for conversion types, which will lead to further selections. The dimension panel and the variable panel display dimension and variable information, respectively.

When NC2Dat is chosen, a GetInfo() function will extract all the information of a NetCDF file. Important information will be directly displayed in the window's related fields. Complete information can be displayed by clicking the More button. When Dat2NC is chosen, the user is assumed to provide the interested variable and its dimensions, as well as other information such as the missing value the scale factor. Information provided from the dialog window is limited. The user can add more attributes to a NetCDF file using the background COM function instead of the dialog window. When Dat2Dat is chosen, the user can either change data types between six options (unsigned char, signed char, short, int, float, double) or change geo-formats between three options (BIP, BIL, BSQ). With a combination of using these functions, the user can freely change data between many sizes, types and formats. In addition, there is an ASCII file conversion that provides several kinds of ASCII to binary data conversions.

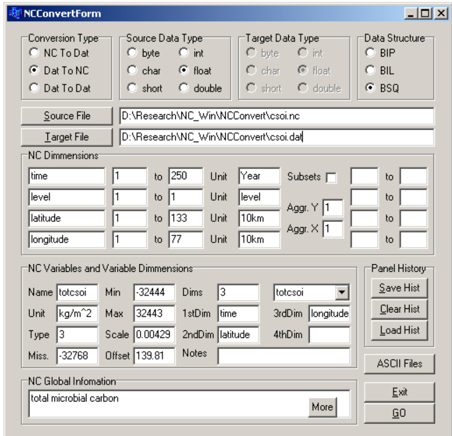

Figure 3: NCWin file conversion window.

Data visualization component

The NetCDF map window (Figure 4a) can display 2-3D data as maps. The 4th D display is under development. Some combo boxes and speed buttons on the tool bar can control the display such as those for selecting the animation speed and changing the display direction (forward and backward). The status bar displays the cursor location and the variable value at the location. A simple map legend can be placed on the right side of the window, which can be automatic or manual. Double clicking an on-site map pixel pops up a trend display window. The curve can display up to nine curves with their location displayed on the legend (Figure 4b).

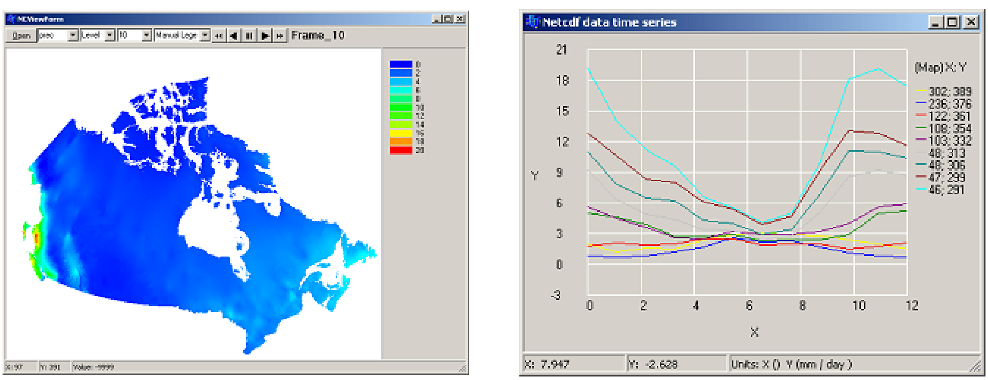

Figure 4: left (a) NCWin map window; right (b) NCWin trends window.

Using NCWin within MS PowerPoint

As mentioned before COM based NCWin can be called by other programs. Because Microsoft Power Point can be programmed with Visual Basic, a Power Point presentation can include a map animation with NCWin. The following example (Figure 5) shows a way of using the map display component (NCMap):

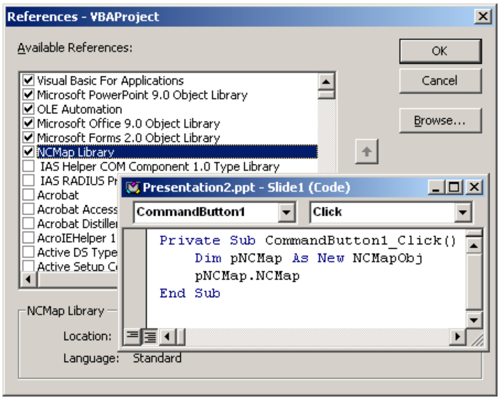

Figure 5: Invoking NCMap COM Server in Visual Basic.

(1) Put a control button (CommandButton1) onto a Power Point slide and double click it to go into the VBA programming mode. (2) Right click on the coding area or press F2 to bring up the Object browser and right click again to open the VB References Dialog. Find the COM library NCMap, select it, and click the OK button. Then a COM object NCMapObj will be available for use. (3) In the CommandButton1_Click() subroutine put two lines as shown in Figure 5. Compile and save. (4) When in presentation mode, a click on the button will bring the NCMap window and shows the map animation as desired. After closing the NCWin, the Power Point presentation continues on.

CONCLUSIONS

Although using NetCDF in environmental modelling is optional to modelers, applications such as NCWin will attract more modelers to the use of NetCDF. Changing a model's data interface to NetCDF is not more difficult than what we did in the InTEC model (Chen et al., 2000), where an old set of 1 km x 1 km resolution, 100 year climate coverages was replaced with a new set of 10 km x 10 km NetCDF maps, with only hundreds of lines of code modification. Then it gives the benefit of sharing and viewing the input and output data at a great convenience. However, calling a COM function is usually slower than calling an internal function defined by a modeler's specific programming language. There are different NetCDF libraries for Java, Fortran, Visual C++ and Borland C++ that can be considered by users. NCWin's language-neutral modelling interface is mainly for model reuse where the user does not need to program NetCDF functions or the user likes to use NCWin's macro modelling functions instead of the original Unidata NetCDF functions. As an application, the NCWin should be considered a developing project. More features can be added to this application such as map enlargement, more color schemes, and additional statistical functions. NCWin is currently built as an in-process COM. But creating an out-process COM that can be used across different machines is just a step forward in the future. Potential users should note that the Unidata NetCDF library is subject to change. Li et al (2003) developed a parallel NetCDF library that advanced the old serial NetCDF file processing technology and made NetCDF more efficient, powerful and flexible.

ACKNOWLEDGEMENT

We gratefully acknowledge the financial supports for this study that was provided by the University of Toronto, the Canadian Forest Service, and the Research Associateship

Programs of the U.S. National Research Council via an agreement with U.S. Geological Survey. We thank Marty Thorne for his early contributions on the prototype of map display and the basic 32-bit RGB color algorithm.

REFERENCES

Benestad, R.E., 2000: *Fifteen Global Climate Scenarios: Conversion to netCDF and Inter-comparison*. DNMI-Report 16/00 KLIMA, pp. 35.

Berry, J.A., Collatz, G.J., Denning, A.S., Colello, G.D., Fu, W., Grivet, C., Sellers, P. J. and Randall, D.A., 1997: SiB2, a model for simulation of biological processes within a climate model. In: Gardingen, P. van, Moody, G. and Curran. P. (Eds.), *Scaling Up*, Society for Experimental Biology, Cambridge University Press, 347-370.

Chen, J.M., Chen, W., Liu, J. and Cihlar, J., 2000: Annual carbon balance of Canada's forests during 1895–1996. *Global Biogeochemical Cycles* 14 (3), 839–849.

Foley, J.A., Prentice, I.C., Ramankutty, N., Levis, S., Pollard, D., Sitch, S. and Haxeltine, A., 1996: An integrated biosphere model of land surface processes, terrestrial carbon balance, and vegetation dynamics. *Global Biogeochemical Cycles* 10(4), 603-628.

Kittel, T.G.F., Royle, J.A., Daly, C., Rosenbloom, N.A., Gibson, W.P., Fisher, H.H., Schimel, D.S., Berliner, L.M. and VEMAP2 Participants, 1997: A gridded historical (1895-1993) bioclimate dataset for the conterminous United States. In: *Proceedings of the 10th Conference on Applied Climatology*, 20-24 October 1997, Reno, NV. American Meteorological Society, Boston.

Li, J., Liao, W., Choudhary, A., Ross, R., Thakur, R., Gropp, W., Latham, R., Siegel, A., Gallagher, B. and Zingale, M., 2003: Parallel netCDF: A Scientific High-Performance I/O Interface. In: *Proceedings of Supercomputing Conference*.

Liu, J.X., Peng, C.H., Dang, Q.L., Apps M.J. and Jiang H. 2002: A component object model strategy for reusing ecosystem models. *Computers and Electronics in Agriculture* 35, 17–33.

Parton, W.J., Schimel, D.S., Cole, C.V. and Ojima. D.S., 1987: Analysis of factors controlling soil organic matter levels in Great Plains grasslands. *Soil Science Society of America Journal* 51, 1173-1179. 465

Potter, W.D., Liu, S., Deng, X. and Rauscher, H.M., 2000: Using DCOM to support interoperability in forest ecosystem management decision support systems. *Comp. Electron. Agric*. 27, 335–354.

Rew, R.K. and Davis, G.P., 1990: The Unidata netCDF: software for scientific data access. *Proceedings, Sixth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, February, Anaheim, California.

Rew, R.K. and Davis, G.P., 1997: Unidata's netCDF interface for data access: status and plans. *Proceedings, Thirteenth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology*, February, Long Beach, California.

Reisdorph, K., 1999: *BORLAND C++ BUILDER 4 Unleashed*. SAMS Publishing, p. 1223.

Rogerson, D., 1997: *Inside COM*. Microsoft Press, Redmond, WA, p. 376.

Smith, W.R., 1997: Model reuse and integration. In: *Proceedings from the IUFRO Conference: Empirical and Process-Based Models for Forest Tree and Stand Growth Simulation*, 21–27 September, 1997, Oerias, Portugal.